Amendments to the Claims

1. (Currently Amended) A distributed information processing system, comprising:

a client device interface processor adapted to receive requests for a type of information from a plurality of remote devices;

a stateless module manager processor adapted to receive and route said requests from said client device interface processor; and

a plurality of information module processors, wherein said information module processors register with said stateless module manager processor, and wherein the stateless module manager processor routes said requests to an appropriate one of said plurality of information module processors in accordance with the type of information requested, wherein the stateless module manager processor handles service collisions in which plural information module processors are capable of responding to the requests, such that only one information module processor processes the requests, wherein the stateless module manager processor enables one of the information module processors to claim the requests and to own subsequent~~the~~ requests ~~afterwards~~based on the type of information being common to each of the requests and the subsequent requests; and

wherein said client device interface processor is adapted to receive a plurality of request types, said request types comprising:

on-demand requests, which are sent to said client device interface processor by a user of one of said remote devices when said user desires an on-demand response;

scheduled requests, which are sent to said client device interface processor by said user when said user desires a plurality of scheduled responses from a subscription service provided by one of said information module processors; and event driven requests, which are sent to said client device interface processor from one of said remote devices when certain criteria are met.

2. (Currently Amended) The distributed information processing system as recited in claim 1, wherein the requests to the device interface processor are formatted as an HTML or plain-text formatted e-mail or serializable Java objects.

3. (Currently Amended) The distributed information processing system as recited in claim 1, wherein the appropriate one of said plurality of information module processors generates a response that is returned to said stateless module manager processor, and wherein said stateless module manager processor routes said response to said client interface device processor for delivery to a requestor.

4. (Currently Amended) The distributed information processing system as recited in claim 1, wherein the stateless module manager processor enables the one of the information module processors to own the subsequent requests independent of which of the plurality of remote devices transmits the requests and the subsequent requests ~~said requests and responses are formatted as serializable Java objects~~.

5. (Currently Amended) The distributed information processing system as recited in claim 1, wherein said requests are made to said stateless module manager processor as one of a synchronous or asynchronous request, wherein synchronous requests are handled on a first-in-first-out basis, and wherein asynchronous requests are processed and returned when completed.

6. (Currently Amended) The distributed information processing system as recited in claim 1, wherein instances ~~of~~associated with said stateless module manager processor are created each time a new request is received and discarded after the request has been handled.

7. (Currently Amended) The distributed information processing system as recited in claim 6, wherein instances associated with ~~of~~ said stateless module manager processor are stateless and multi-threaded.

8. (Currently Amended) The distributed information processing system as recited in claim 1, wherein information module processors are loaded locally and remotely, wherein local module processors reside on a same physical device as said stateless module manager processor, and wherein remote module processors are located on other devices.

9. (Currently Amended) The distributed information processing system as recited in claim 8, wherein communication between locally loaded module processors and said stateless module manager processor is accomplished via memory calls, object inheritance or inter-process communication.

10. (Currently Amended) The distributed information processing system as recited in claim 8, wherein communication between remotely loaded module processors and said stateless module manager processor is accomplished via TCP/IP sockets.

11. (Currently Amended) The distributed information processing system as recited in claim 1, wherein the subscription service

further comprises a subscriber database, wherein information is sent by said information module processors, and said subscriber database is consulted to determine to which users of said remote devices the information should be forwarded.

12. (Currently Amended) A method of receiving and responding to requests for electronic information in a distributed information processing system, the method comprising:

receiving requests for a type of information at a client device interface;

forwarding said requests to a stateless module manager;

consulting a registry of available information modules; and

forwarding said requests to an appropriate information module as determined in accordance with the type of information requested;

handling service collisions if plural information modules are capable of responding to the requests, such that only one information module processes the requests, and enabling one of the information modules to claim the requests and to own subsequent~~the~~ requests ~~afterwards~~based on the type of information being common to each of the requests and the subsequent requests;

wherein said client device interface is adapted to receive a plurality of request types, said request types comprising:

on-demand requests, which are sent to said client device interface by a user of one of said remote devices when said user desires an on-demand response;

scheduled requests, which are sent to said client device interface by said user when said user desires a plurality of scheduled responses from a subscription service provided by one of said information modules; and

event driven requests, which are sent to said client device interface from one of said remote devices when certain criteria are met.

13. (Previously Presented) The method of claim 12, further comprising:

maintaining a list of supported services provided by each of said information modules; and

registering said information modules for responding to requests for said type of electronic information.

14. (Currently Amended) The method of claim 12, wherein the one of the information modules owns the subsequent requests independent of a source of the requests and the subsequent requests~~wherein said requests and responses are formatted as serializable Java objects~~.

15. (Previously Presented) The method of claim 12, wherein said requests are made to said stateless module manager as one of a synchronous or asynchronous request, wherein synchronous requests are handled on a first-in first-out basis, and wherein asynchronous requests are processed and returned when completed.

16. (Previously Presented) The method of claim 12, said method further comprising:

creating an instance of said stateless module manager upon receiving said request; and

discarding said instance after said response has been handled.

17. (Currently Amended) A computer readable medium containing computer executable instructions for receiving and responding to requests for electronic information in a distributed information processing system, said computer executable instructions for performing the steps of:

receiving requests for a type of electronic information at a client device interface;

forwarding said requests to a stateless module manager;

consulting a registry of available information modules;

forwarding said request to an appropriate information module as determined in accordance with the type of electronic information requested;

handling service collisions if plural information modules are capable of responding to the requests, such that only one information module processes the requests, and enabling one of the information modules to claim the requests and to own ~~the~~ subsequent requests ~~afterwards~~ based on the type of electronic information being common to each of the requests and the subsequent requests;

wherein said client device interface is adapted to receive a plurality of request types, said request types comprising:

on-demand requests, which are sent to said client device interface by a user of one of said remote devices when said user desires an on-demand response;

scheduled requests, which are sent to said client device interface by said user when said user desires a plurality of scheduled responses from a subscription service provided by one of said information modules; and

event driven requests, which are sent to said client device interface from one of said remote devices when certain criteria are met.

18. (Previously Presented) The computer readable medium of claim 17, further comprising computer executable instructions for performing the steps of:

maintaining a list of supported services provided by each of said information modules.

19. (Currently Amended) The computer readable medium of claim 17, wherein the one of the information modules owns the subsequent requests independent of a source of the requests and the subsequent requests ~~wherein said requests and responses are formatted as serializable Java objects~~.

20. (Previously Presented) The computer readable medium of claim 17, wherein said requests are made to said stateless module manager as one of a synchronous or asynchronous request, wherein synchronous requests are handled on a first-in-first-out basis, and wherein asynchronous requests are processed and returned when completed.

21. (Previously Presented) The computer readable medium of claim 17, further comprising executable instructions for performing the steps of:

creating an instance of said stateless module manager upon receiving said request; and

discarding said instance after said response has been handled.

22. (Currently Amended) A stateless module manager processor that manages requests for electronic information received at a mailbox, comprising:

a registry of information module processors;

a module loading function processor for dynamically loading said information module processors upon receipt of said request for electronic information, wherein said requests are made as one of a serializable Java object, XML placed in an HTTP header, or an XML-RPC-enabled web server, wherein said requests are either synchronous or asynchronous, wherein synchronous requests are handled on a first-in-first-out basis, and wherein asynchronous requests are processed and responses returned in accordance with processing times of the request;

wherein said stateless module manager processor routes said requests to an appropriate information module processor for resolution, and wherein said appropriate information module processor resolves said requests and returns responses to said stateless module manager processor;

wherein said stateless module manager processor maintains a list of supported services provided by each of said information module processors and handles service collisions such that if plural information module processors register as supporting a same service related to requests, the stateless module manager processor determines one of said plural information module processors to handle said requests by enabling the one information module processor to claim the requests and to own subsequent~~the~~ requests ~~afterwards~~independent of a source of the requests and the subsequent requests;

wherein instances of said stateless module manger processor are created each time a new request is received and discarded after the request has been handled;

wherein said stateless module loading function processor includes local and remote module loading function processors, wherein said local loading function processor loads information module processors that reside on a same physical device as said stateless module manager processor, wherein said remote loading function processor loads information module processors that reside on devices logically connected to said stateless module manager processor, wherein said local module processors communicate with said stateless module manager processor via one of memory calls, object inheritance, and inter-process communication, and wherein said remote information module processors communicate with said stateless module manager processor via TCP/IP sockets; and

further comprising a user interface, wherein said user interface is adapted to configure said stateless module manager processor; and

wherein said stateless module manager processor is adapted to receive a plurality of request types, said request types comprising:

on-demand requests, which are sent by a user of one of said remote devices when said user desires an on-demand response;

scheduled requests, which are sent by said user when said user desires a plurality of scheduled responses from a subscription service provided by one of said information module processors; and

event driven requests, which are sent from one of said remote devices when certain criteria are met.

23-31. (Cancelled).

32. (New) The stateless module manager processor of claim 22, wherein the one information module processor owns the subsequent requests based on the type of information being common to each of the requests and the subsequent requests.